How to start with RISM-MOL

Return to the main page
Download HowTo in a single pdf document

What is RISM-MOL

What is RISM-MOL?
What is RISM?
What is infinitely diluted aquoeus solution?
How to install RISM-MOL?
How should I start RISM-MOL?

How to work with RISM-MOL

How do I describe my solute molecule?
How to create the input file?
How to calculate Hydration Free Energy?
How to display calculation results?
How to save results of the calculations?
What Hydration Free Energy expression should I use?

How to calculate site-site RDF functions?
Example of calculating RDF functions
How to use RDF in external programs?

How to calculate solutions gamma, c of RISM equations?
Example of working with gamma, c , and potLong parameters

How can I use different inpiut and output units?

Extra parameters

Extra parameters of startRISM function
How can I choose closure relation?
How to use repulsive bridge?

How to change the accuracy of calculations?

What is RISM-MOL?

RISM-MOL is the multigrid solver of the RISM equations for the aqueous infinitely diluted solution. RISM-MOL is written in matlab and thus it is crossplatform.

- In the current version the temperature and the density are fixed to be T=300K and n=33.7 water molecules per nm^3 respectively.
- RISM-MOL allow to calculate Hydration Free energies of an arbitrary solute molecule, by six different expressions (HNC,HNCB,GF,PW,PWC,KH) (see section *HFE expressions* in the manual for their description).
- Also RISM-MOL calculates the solute excluded volume.
- As well, with RISM-MOL you may find the site-site Radial Distribution functions between the solute atoms and the Water Oxygen and Hydrogen.
- RISM-MOL have an flexible settings, so you may change force-field parameters of solute and water, water model, closure relations, accuracy of calculations.

What is RISM?

RISM is model in the Integral Equation Therory of Liquids, which describes the solvent structure in therms of continious site-site correlation functions. It allows to calculate atom-atom Radial Distribution functions (RDF). There are expressions, which allow to calculate Hydration Free Energy with the RISM theory. Read the User Manual or special literature for details.

What is infinitely diluted aquoeus solution?

Infinitely diluted aqueous solution is a model, which offently used for Hydration Free Energy calculations. It is the model of the single solute molecule, surraunded by a vaste of solvent (water) molecules.

How to install RISM-MOL?

To install RISM-MOL you should:

- Dowload license agreement form the RISM-MOL site: http://compchemmpi.wikispaces.com/file/view/LicenseAgreementRISM_MOL.pdf
- Print and sign license agreement
- Scan signed license agreement and send to address:
- If you request is approved, the RISM-MOL package and installation instructions will be mailed to you

How should I start RISM-MOL?

The interface with a program is made by the only matlab function - startRISM. If you have setup the paths to the program properly, you can run this function form the Matlab command window

Function takes as input the coordinates and force-filed parameters of the solute molecule form the Input solute file (see How do I describe my solute molecule?) and calculates hydration free energies (HFE) by different expressions,(see section "Hydration Free Energy Expressions" in the User Manual), Excluded volume , site-site Radial Distribution Functions, solutions of the RISM equations gamma and c and long-range Coulomb potential.

To calculate the Hydration Free Energies using the startRISM function use the following command in Matlab Command Window:

HFE = startRISM(<input file name >);

For example, whith a program is provided the test input file 'methane.txt'. Then you may use:

HFE = startRISM('methane.txt');

HFE is a structure, which contains Hydration Free Energies calculated by different methods.(See How to calculate Hydration Free Energy? for details).

If you use to calculate some more parameters, you should use the full function syntax:

[HFE,g,grid,gamma,c,potLong] = startRISM (fname, input_units, output_units,xparam)

Here HFE, g, grid, gamma, c, potLong are output parameters fname, input_units, output_units, xparam are input parameters

For Exmple you may use such command:

[HFE, g,grid,gamma, c,potLong] = startRISM ('methane.txt', ...
struct('Distance','Angstr','Energy','kcal/mol'), ...
struct('Distance','Angstr','Energy','kcal/mol'), ...
'user_LambdaCoupling=0.5;')

You may also omit last some input argument, to use the default parameters, and some output arguments, if you don't need them:

[HFE,g,grid] = startRISM ('methane.txt')

BUT BE CAREFUL The order of parameters **should be the same** (so to **CANNOT** use [g,grid] = startRISM ('methane.txt'), because parameter should be all the time the first)

The meaning of the parameters, and references to their detailed description:

Input parameters

- fname - path to the input file with the solute structure. see How do I describe my solute

molecule?

- input_units - units, used for the coordinates in the structure file and for sigma LJ parameters (by default - Angstroms). See How can I use different inpiut and output units? .
- output_units - units, used for calculated Hydration Free Energies and distance vector grid. See How can I use different inpiut and output units? .
- xparam - string, containing extra parameters. Any parameters from *UserParameters.m* file can be used here. See Extra parameters of startRISM function

Output parameters

- HFE - structure, which contains calculated hydration free energies and excluded volume. See How to calculate Hydration Free Energy? for details.
- g - site-site radial distribution functions. See How to calculate site-site RDF functions? .
- grid - grid vector. See How to calculate site-site RDF functions? .
- gamma,c - short-range solutions of the RISM equations. See How to calculate solutions gamma, c of RISM equations?
- potLong - long range part of potential. See Example of working with gamma, c , and potLong parameters

How do I describe my solute molecule?

Input data for the solute is stored in a separate file. Structure file is an regular text file. There should be six space or tab separated columns of numbers in it. Number of rows is equal to the number of atoms in the solute molecule. Each row corresponds to the one atom.
Data columns are: **X Y Z Sigma Epsilon Charge**

- **X,Y,Z** are coordinates of the atom
- **Sigma** and **Epsilon** are Lenard-Johnes potential parameters for the atom
- **Charge** is the charge of the atom

X Y Z and **Sigma** should be given in the input distance units.
Epsilon should be given in the input Energy units.
Charge should be given in the atomic charge units (1 a.u. charge = charge of proton).

By default input and output distance units are angstroms and energy units **kcal/mol**. However there is possibility to write an input file in a different units and specify these units as parameters of the function startRISM.

How to create the input file?

You may create input file in any text editor you want. For example, in matlab you may type the following command:

edit input.txt

Then in the editor you may describe your solute. For example, for the simple Argon-like atom you may type in the file input.txt:

0 0 0 3.5 0.15 0

By default, units are Angstroms and kcal/mol.
So, this file will describe a molecule, which is placed at position (0,0,0), which have the Sigma Lenneard Jones parameter sigme=3.5 Angstr, Epsilon LJ parameter Epsilon=0.15 kcal/mol, and have a zero charge.
For example of more complicated system (methane molecule), you may refer the input file methane.txt, which is distributed with the program. To see it use the command:

edit methane.txt

How to calculate Hydration Free Energy and excluded volume?

To calculate Hydration Free Energy you may use a command:

HFE = startRISM(<input file name>);

With the RISM-MOL is provided the file methane.txt, which contain the structure of the methane molecule. Then, to calculate the Hydration Free Energy of methane, you may use a command:

HFE = startRISM('methane.txt');

In that case HFE will contain values of the chemical potentials, calculated by different expressions, and the value of the excluded volume. HFE is a structure with such fields:

- HFE.HNC -- Hydration Free Energy, calculated using Hyper-Netted-Chain assumption (HNC)
- HFE.G -- Hydration Free Energy, calculated using Gaussian Fluctuations expression (GF)
- HFE.HNCB -- Hydration Free Energy, calculated using the HNCB expression. (HNCB)
- HFE.PW -- Hydration Free Energy, calculated using Partial Wave expression (PW)
- HFE.PWC -- Hydration Free Energy, calculated using Partial Wave expression with excluded volume correction
- HFE.VUA -- excluded volume
- HFE.KH -- Hydration Free Energy, calculated using Kovalenko-Hirata expression for PLHNC closure

Hydration Free Energies HFE.HNC,HFE.G,HFE.HNCB,HFE.PW,HFE.PWC,HFE.KH are in the output_units.Energy units (by default: kcal/mol),

Excluded volume HFE.VUA is in output_units.Distance$_3$ units (by default Angstr$_3$)

How to display calculation results?

Output parameters of the startRISM are Matlab matrices and structures. To display them you may use matlab command *disp*. For example, to show Hydration Free Energies you may use:

```
HFE=startRISM('methane.txt');
disp(HFE);
```

If you need some particular expression, you may use syntax: HFE.< expression name > For example, to show results of PW Hydration free energy calculations, you may use:

```
disp(HFE.PW);
```

How to save results of the calculations?

To save results of calculations, you may use the matlab command *save <file name> <variable name>* For example, to save results of hydration free energy calculations, use:

```
HFE=startRISM('methane.txt'); save HFE.mat HFE
```

The file HFE.mat will be created. When you will start matlab next time, you may use command *load < filename >* For example:

```
load 'HFE.mat' disp(HFE);
```

The same applied not only to the Hydration free energies, but for all output parameters of the startRISM function. For example to save and re-load RDF functions and grid use:

```
[HFE,g,grid]=startRISM('methane.txt');
save g.mat g
save grid.mat grid
```

(now you may restart matlab safely: the data is saved) To load data:

```
load grid.mat load g.mat
```

For more information about saving the data in matlab, refer matlab help on function *save*. It may be done by typing the command:

```
doc save
```

What Hydration Free Energy expression should I use?

There are no universal rules, how to choose the best expression, all depends on what you need. From our experience we can say, that GF and PW expressions give much better correspondance to the experiment.
Also you should be careful while using HNC,HNCB and KH expressions, because for each of these expressions there is special closure relation (see the section *HFE expressions* in the User Manual)

How to calculate site-site RDF functions?

To calculate RDF functions use the command:

[HFE,g,grid] = startRISM(<input file name>);

For example, to calculate methane-water site-site RDF functions use the command

[HFE,g,grid] = startRISM('methane.txt');

Parameter g contains site-site Radial Distribution Functions , calculated by RISM algorithm. The radial distribution functions are stored in the matrix. Each column of the matrix contain one distribution function.

Odd columns (1,3,5,...) contain the distribution functions between the solute sites and water oxygen. Column with number 2k+1 contain the distribution functions between the k-th solute atom and water oxygen (solute atoms are numbered in the same order, as they give in the Solute input file. See esection How do I describe my solute molecule for details.

Even columns (2,4,6,...) contain the distribution functions between the solute sites and water hydrogen. Column with number $2k+2$ contain the distribution functions between the k-th solute atom and water hydrogen (solute atoms are numbered in the same order, as they give in the \emph{Solute Input File}.

You may access particular RDF function by using the matlab colon operator ':'
For example g(:,3) will return the RDF function between the 2-ns solute site and water oxygen (because 3=2*1+1)

grid is the column-array, containing the samples of distance from the solute atom. That samples corresponds to the samples in the produced RDF functions (columns of *g* array). g(i,k) is the value of the k-th RDF function at the distance grid(i).

Example of calculating RDF functions

Let for example we would like to calculate site-site RDF functions between the atoms of methane molecule and water. We may calculate them by typing such line in the Matlab Command Window:

```
[HFE,g,grid]=startRISM('methane.txt')
```

To access particular RDF function we use the matlab colon operator ':'.
The Input file for the methane molecule you may find in the 'methane.txt' file, which is distributed with the RISM-MOL. In that file the first line corresponds to the Carbon atom, other lines correspond to hydrogen. Then, to plot the Carbon-Water oxygen and Carbon-water hydrogen RDF functions we may use the following commands:

```
plot(grid,g(:,1),grid,g(:,2))
```

To make them more recognizable we use additional visualization commands:

```
legend('C-O','C-H');
xlim([0 10]);
xlabel('Angstroms');
ylabel('g(r)');
```

More information about the visualization abilities of matlab you may find using the matlab help:

```
doc plot
```

How to use RDF in external programs?

To use calculated data in the other programs, you should store them in the ASCII format, which is readable by many graph editors. It may be done by using the *save* command with key *ASCII* .

```
save g.txt g -ASCII
```

The file will contain the olumns with all g(r) functions in usual human-readable text format. Ofently, visualization program require, that *grid* vector should be the first column. To do it, you may use the matlab matrix concatanation operators ([]):

```
A=[grid g];
save out.txt A -ASCII
```

In that case the file, there the first column is grid and all other columnsare RDFs will be produced. For more details, you may refer the matlab documentation:

```
doc paren
```

How to calculate solutions gamma, c of RISM equations?

To obtain the solutions of the RISM equations you may use the gamma,c and potLong parameters of startRISM function.

[HFE,g,r,gamma,c,potLong]=startRISM(<input file>);

gamma is the matrix, containing the short-range site-site indirect correlation functions
c is the matrix, containing the short-range site-site direct functions
potLong is the matrix, containing the long-range potential
The format of the matrices as the same, as for the site-site RDF functions (see How to calculate site-site RDF functions?) (See User Manual for more detailed explanations)
To obtain full direct and indirect correlation functions gammaFull and cFull one may use equations:

gammaFull = gamma + potLong
cFull = cFull - potLong

Example of working with gamma, c , and potLong parameters

To obtain the solutions of the RISM equations for methane molecule you may use the following script:

[HFE,g,grid,gammaS,cS,potLong]=startRISM('methane.txt');

To plot short-range functions you may use the script

figure,plot(grid,gammaS), xlim([0 10]), title('Gamma functions')
figure, plot(grid,cS), xlim([0 10]), title('c functions')

To plot the long-range potential you may use:

figure,plot(grid,potLong), xlim([0 10]), title('\beta u_{Long}')

To find full functions you may use the script:

cFull = cS - potLong; gammaFull = gammaS + potLong;

To plot the full functions you may use the script:

figure,plot(grid,gammaFull), xlim([0 10]), title('Gamma functions')
figure, plot(grid,cFull), xlim([0 10]), title('c functions')

How can I use different inpiut and output units?

To define the input/output parameters type in matlab command window:

inputUnits = struct('Distance',<distance unit name>,'Energy',<energy unit name>)

Example:

```
inputUnits = struct('Distance','Angstr','Energy','kcal/mol')
outputUnits = struct('Distance','Angstr','Energy','kcal/mol')
HFE=startRISM('methane.txt',inputUnits,outputUnits)
```

Supported units

Distance Units

- `m' -- meters
- 'Angstr' -- Angstroms
- 'Bohr' -- Atomic distance units (Bohr)

Energy Units

- 'J' -- joules
- 'Hartree' -- Hartree , atomic energy units
- 'kcal/mol' -- kilocalories per mole.
- 'kJ/mol' -- kilojoules per mole,
- 'eV' -- electron Voltes

Extra parameters of startRISM function

Sometimes you need to change some parameters of the algorithm (e.g closure relation, or accuracy). As was said before, it can be done by editing the *UserParameters.m* file. However, it can be done simplier: in RISM-MOL was implemented the possiblility to change parameters *on fly*, just for the current run of the program. To do it you should use *xparam* parameter of the *startRISM* function. *xparam* is the string containing the values of the algorithm parameters, separated by semicolon (';'). Parameters are given in the format

Parameter name = Parameter value

For the list of avalilable parameters you may refer the section User Parameters in the User Manual (parameters are the same, which are used in the *UserParameters.m* file).

Note If the parameter value is string, you should use two **single** quotes instead each quote. For example: to change the closure to HNC use the folowing value of xparam:

```
xparam='user_Closure="HNC";'
```

Example of using extra parameters

For example, you need to run the RISM progam for the methane molecule, HNC closure, and to prevent the divergence you would like to use coupling parameter user_LambdaCoupling=0.05 You

may do it by typing following commands in the matlab command window:

inputParameters = struct('Distance','Angstr','Energy','kcal/mol') outputParameters = struct('Distance','Angstr','Energy','kcal/mol')
xparam='user_Closure="HNC";user_LambdaCoupling=0.05;'
HFE=startRISM('methane.txt',inputParameters,outputParameters,xparam)

How can I choose closure relation?

By default Partially-Linearized closure PLHNC is used. If you want to use HNC closure, specify it in the last argument of startRISM function:

HFE = startRISM(< input file >, < input units >, < output units >, 'user_Closure="HNC";');

For example, for methane molecule you may use following script:

Units = struct('Distance','Angstr','Energy','kcal/mol');
HFE=startRISM('methane.txt',Units,Units,'user_Closure="HNC"');

How to use repulsive bridge?

If you want to use Repulsive Bridge, specify it in the last argument of startRISM function:

HFE = startRISM(< input file >, < input units >, < output units >,
'user_UseRepulsiveBridge="yes";');

If you want to use Repulsive Bridge with HNC closure use the Command:

HFE = startRISM(< input file >, < input units >, < output units >,
'user_UseRepulsiveBridge="yes";user_Closure="HNC"');

For example, for methane molecule you may use following script:

Units = struct('Distance','Angstr','Energy','kcal/mol');
HFE=startRISM('methane.txt',Units,Units,'user_Closure="HNC";user_UseRepulsiveBridge="yes"');

How to change the accuracy of calculations?

If you want to change the accuracy of calculations, specify it in the last argument of startRISM function:

HFE = startRISM(< input file >, < input units >, < output units >,
'user_AccuracyThreshold=<Accuracy Threshold>');

The less is Accuracy Threshold, the better is solution.
Default value is user_AccuracyThreshold=1e-4
For example, to change accuracy of calculations for methane molecule you may use following script:

```
Units = struct('Distance','Angstr','Energy','kcal/mol');
HFE=startRISM('methane.txt',Units,Units,'user_AccuracyThreshold=1e-5;');
```

What should I do if iterations diverge?

If iterations diverge, they either will return NaN instead of HFE values, or just startRISM script will not stop.
If the script do-not finish for a very long time, you may terminate it Typing Ctrl+C If the iterations diverge, you may try to use another (smaller) value of the coupling parameter user_LambdaCoupling. Default value is user_LambdaCoupling=0.5, which is ok for PLHNC closure. However, for HNC closure and for HNCB corrections you may need to set it smaller, because otherwise iterations may diverge. You may change this parameter by specifying it in the last argument of startRISM function:

```
HFE = startRISM(< input file >, < input units >, < output units >,
'user_LambdaCoupling=<Lambda Value>');
```

For example, for methane molecule you may use:

```
Units = struct('Distance','Angstr','Energy','kcal/mol');
HFE=startRISM('methane.txt',Units,Units,'user_LambdaCoupling=0.1;');
```